0300

#4

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of
JOHAN C. TALSTRA ET AL

Atty. Docket No.
PHN 17,410

Serial No.:09/548,727

Group Art Unit:

Filed: APRIL 13, 2000

Title: METHOD AND SYSTEM OF COPY PROTECTION OF INFORMATION

Honorable Commissioner of Patent and Trademarks
Washington, D.C. 20231

CLAIM FOR PRIORITY

Sir:

A certified copy of the European Application No. 99302887.7 filed April 14, 2000 and referred to in the Declaration of the above-identified application is attached herewith.

Applicants claim the benefit of the filing date of said European application.

Respectfully submitted,

June 27, 2000
Enclosure

By Michael E Belk
Michael E. Belk, Reg. 33,357
Attorney
(914) 333-9643

CERTIFICATE OF MAILING

It is hereby certified that this correspondence is being deposited with the United States Postal Service as first-class mail in an envelope addressed to:
COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

On 6/28/00

By Michael E Belk

S:\BE\PRIORITY.DOC

PHN 17410
US

Europäisches Patentamt | European Patent Office | Office européen des brevets

Bescheinigung | Certificate | Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. | Patent application No. | Demande de brevet n°

99302887.7

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

I.L.C. HATTEN-HECKMAN

DEN HAAG, DEN
THE HAGUE,
LA HAYE, LE 19/04/00

C. PASTUREL

EPA/EPO/OEB Form 1014 - 02.91

Europäisches Patentamt
European Patent Office
Office européen des brevets

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.:
Demande n°: 99302887.7

Anmeldetag:
Date of filing:
Date de dépôt: 14/04/99

Anmelder:
Applicant(s):
Demandeur(s):
Koninklijke Philips Electronics N.V.
5621 BA Eindhoven
NETHERLANDS

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
Secure means for communicating watermark related copy-protection information in a PC

In Anspruch genommene Priorität(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat: State: Pays:	Tag: Date: Date:	Aktenzeichen: File no. Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:
/

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

Secure means for communicating watermark related copy-protection information in a PC.

Introduction

The invention relates to an arrangement for receiving via a transfer signal encoded content information and supplemental information, which content information comprises a watermark at least partly representing the supplemental information, the arrangement comprising a receiver device for receiving the transfer signal, a detector for detecting watermark information in dependence on the watermark, and a decoder coupled to an output of the receiver device for decoding the content information, which receiver device comprises control means for controlling the reproduction of the content information in dependence on the supplemental information.

Such a transfer system is known from WO 97/13248 (PHN 15391). In the transfer system information is transferred from the transmitter via a transfer signal to a receiver device, e.g. from a video producer via an optical disc to a disc drive for playback. The document describes that video and audio content is increasingly transmitted and recorded in a digitally encoded form, for example, an MPEG bitstream. There is a growing need to transfer supplemental information logically related to the content information, which supplemental information is intended for controlling the reproduction of the content information. The supplemental information may comprise information on the rights of the owner or originator of the content information. For example a marker is to be accommodated in such an encoded signal so as to classify the encoded signal as authentic program material. Marking digital signals is particularly useful in copy protection applications, wherein the supplemental information indicates the copyright status. Therefore the supplemental information should be protected against manipulation. The mark, also referred to as watermark, can effectively take the form of a multi-bit watermark pattern representing some supplemental information, e.g. indicating that the encoded signal constitutes copy protected content. In a digital video system, e.g. based on the digital videodisc (DVD), copy control can be based on detection of electronic watermarking. Watermarks are minor, imperceptible modifications to the video, which can be

detected electronically. Such watermarks can be resistant to typical signal processing, including format conversions (e.g. PAL to NTSC), and can be detected to retrieve copyright information about the video. Watermarks are used for playback control. The basic idea of playback control is that any drive refuses to pass video content if that content contains a watermark that classifies the video as being no-copy while that video is found on a recordable medium. Hence playback control requires detection of the watermark within the drive, and a detector should be on the same chip as the drive control electronics or on the same circuit board in the drive. Noise-like, pixel-domain watermarks are not suitable for detection by a detector in the receiver device, because the complexity of the detector has to remain below a few thousand gates, as drives and DVD RAM recorders are designed as simple storage devices without any 'intelligence' to interpret data. Watermark detection would imply that such devices have process the content data, e.g. to demultiplex and interpret MPEG video streams, at least including run-length Huffman decoding of DCT coefficients. Hence a requirement of simplicity of playback control can not effectively be met by pixel-domain watermarks. So the known system has the problem, that the drive must be provided with a complex watermark detector.

Another arrangement for receiving via a transfer signal encoded content information and supplemental information, which content information comprises a watermark at least partly representing the supplemental information, the arrangement comprising a receiver device for receiving the transfer signal, a detector for detecting watermark information in dependence on the watermark, and a decoder coupled to an output of the receiver device for decoding the content information, which receiver device comprises control means for controlling the reproduction of the content information in dependence on the supplemental information, is known from WO99/11064 (PHN 16517). Embodiments relating to this arrangement for receiving via a transfer signal encoded content information and supplemental information can be found in WO99/11064 (PHN 16517).

It is an object of the invention to provide a more flexible system for controlling the playback of content information in dependence of supplemental information.
DVD-Video material is currently protected by the DVD-FORUM Content Scrambling System (CSS). The content providers are looking for ways to enhance this protection system and are

requesting additional layers of protection for their IP. A set of proposals for a new Content Protection System (CPS) is being considered at the CPTWG based on *watermarking* the Video content. (CPTWG = Copy-protection Technical Working Group, a copy-protection discussion/standardization forum of consumer electronics-, IT- and film-industries, convening monthly in Burbank, CA).

This watermark is used to effectuate both *playback-* and *record-control*. Record control implies that a recorder refuses to make a copy of a piece of video that contains an appropriate *copy-never* or *copy-no-more* watermark. (See ref 0,0,[5] for a definition of the terms copy-no-more, copy-never and copy-once). Because it is relatively easier for a pirate to modify his own recorder than the players of the customers to which he will try to sell his counterfeited video-material, perhaps *playback control* is more relevant. Playback control entails only allowing playback of content with a watermark, when the information carrier on which the content resides is of a nature compatible with the watermark. E.g. a movie with a watermark "*copy-never*" should always be on a factory pre-recorded ROM (or "silver") disk. If the movie resides on a recordable ("golden") disk, or a non-authorized "silver" disk, playback should stop. (A more serious form of disk-type distinction is to check whether the pits on the disk do not lie on a regular spiral but on a slightly *wobbled* spiral; upon copying (even bit-copying) this "wobble" is lost). A similar system is envisioned for audio applications such as SACD and DVD-Audio, or other multimedia applications.

From the point of view of implementing playback control in a digital player, the watermark detector will typically be part of a *decoder*; by decoder we mean that part of the player that is used to turn the bits from the information carrier into a visible/audible signal (e.g. an MPEG decoder—soft- or hardware—, and/or D/A-converter). On the other hand, the nature of the information carrier will be determined in the so-called *drive*, which reads the actual bits from tape/disk. The copy-protection information gathered by both pieces of functionality will have to be shared via some *protocol*, in order to effectuate playback control. See Figure 1 for a typical play-control set-up for a video player.

In the sequel, we will use the terms 'decoder' and 'application' interchangeably. From a security standpoint, there is no serious problem in a stand-alone tabletop player, where integration and absence of a well-defined public interface between *drive* and *decoder* present almost insurmountable problems to the average hacker. In a Personal Computer environment however, drive and decoder are usually physically separate entities, connected via an open, well-documented (PCI) bus. Furthermore, they communicate under the guidance of an appropriate software application.

This has two implications:

1. The drive and decoder have to decide independently whether to cease playback, based on the information they obtain from each other.

2. The software application orchestrating the actions of decoder and drive cannot be trusted with this decision, as it is easily replaced by a malevolent version (perhaps downloaded from the internet). Moreover, this malevolent version may actively interfere with the autonomous playback control of drive/decoder by intentionally modifying messages from decoder to drive and vice versa.

To increase the security of the watermarking copy protection system, the watermark is generally chosen to be content dependent (e.g. every movie will have it's own watermark, such that hacking the watermark in one film, doesn't necessarily expose all films), which is to be coupled to an appropriate *physical property* of the disk, the so called *physical mark* or *diskmark*. In other words, the watermark carries a (single or multiple byte) payload, which is related to the payload of the physical disk property. Exchanging these two numbers securely between drive and decoder will be the subject of this invention disclosure.

There exists a fairly easy hack, the so-called the man-in-the-middle attack, see Appendix 0. This attack makes it necessary for the exchange protocol *also* to check whether the content arriving at the decoder is (a subset of) the data transmitted by the drive.
Given the mass-product nature of esp. the drive, this protocol should be as simple as possible, and not interfere with the normal functionality of either drive or decoder.
To summarize, the protocol should be:

I. secure
  a) against man-in-the-middle attack.
  b) against a hacker obtaining watermark/drive payload
II. cheap and simple in both soft- and hardware
III. not impair drive and decoder in their normal functioning
IV. should be compatible with the constraints of existing standard interfaces, protocols and storage formats

In this invention disclosure, we will suggest a protocol that will satisfy the above design constraints. We will give a particular example for the case of DVD-video, which can probably be generalized to audio or other formats multimedia formats.
The three main invention ideas (see section 3 Conclusions for a more general description) are:

1. In general, to continuously exchange between drive and application a *characteristic of the content* or *summary* that is being streamed, to avoid man-in-the-middle attacks.
2. In particular for a DVD-video drive, to use the presence of pack_start_code as the first 4 bytes in a sector, as a criterion for considering that sector to contain video (or more precisely: MPEG Program Stream information)
3. In particular for DVD-video, to base the characteristic/summary of bullet 0 on the SCR_base[]MPEG-field, which can be guaranteed to be transmitted by the drive and received by the application; i.e. not to summarize all data but just the part related to the correct SCR_base[].

Protocol Characteristics

Assumption: because the communication between drive and application is to be tamper-proof, and preferably secret, we assume that through some means, a secret K, not known to the outside world, has been shared between drive and application. Examples are:

1. The bus-key in CSS (Content Scrambling System, data encryption method for DVD-video disk)
2. A universal secret embedded in drive-silicon and application-silicon.
3. a key shared as the result of a to be defined secure authentication protocol (e.g. CSS-2).

Typically K is a 64- or 128-bit number.

Appendix 0 clarifying the Man-in-the-middle attack, concludes the following regarding item 0 0 in section 0: the vulnerability exploited by this attack is that the decoder receives other data than that transferred by the drive. To thwart the attack, in the *copy-control messages* between drive and decoder:

1. the drive needs to report to the decoder a summary of the video that it transmitted,
2. the decoder needs to report to the drive a summary of the video that it received.

This leads to the first invention claim, mentioned in the introduction.

A complication in this scenario is that not all of the data requested from the drive is sent to the decoder: e.g. the table of contents of the disk and other file-management information is read from the disk and processed by the operating system, but not by the application! If the drive were making a summary based on all data transmitted and the decoder only on the data it receives, a *false alarm* would be raised even during legal playback...

It is therefore essential that the summary that is being exchanged concerns *only* that part of the data that will end up in the application! A particular problem is that this stream of summary verification messages has to be synchronized.

From now on we will specialize to the case of DVD-Video, but we believe that generalizations of the protocol that will be described, exist.

For the particular case of DVD-Video, there is a way to construct such a *unique* summary. First off: all data on DVD-disks is divided into blocks of 2048 bytes called *sectors*. Currently, for the outside world to access data on the disk, the only way is to request *entire sectors* at a time 0.[2]. The specification of DVD video is such that video data that will be sent to the application (an MPEG decoder card) is never mixed with "administrative data" (which is not sent to the decoder) in the same sector. The drive has no a priori knowledge, however, to distinguish sectors containing administrative data from those with video.

According to the DVD-Video specification (0 Part 1), the data that will be received by the MPEG-decoder, the MPEG Program Stream stored on a DVD, is organized into a sequence of pack()'s, all with length 2048 bytes. Every pack() is stored in exactly 1 sector on the disk. Therefore the decoder *also* knows about sector-boundaries by identifying pack()'s. A pack()has a structure that can be found in Figure 2.

When a sector starts with a 4-byte pack_start_code, the drive knows that this sector will eventually be received by the decoder. Conversely if the first 4 bytes of the sector do not equal the pack_start_code, the sector is not bound for the decoder, and should be ignored for the "summary" computation. This solves the problem of selecting the right data to compute a summary on, and clarifies the second invention claim in the introduction.

Because it is computationally too intensive to compute, exchange and verify a summary $C(T_K)$ of each sector that the drive transmits, the drive and decoder **should select a few sectors/pack()**s based on their shared secret K. They will compute a unique feature of that sector, and securely exchange that feature together with the watermark/disk-mark information. A pirated software driver in the "Man-in-the-middle" scenario cannot abuse a compliant decoder by occasionally sending it a sector from the drive, and thus generating a valid *Copy Control Message* because (s)he would not know, which sector to send.

In Table 1. the SCR_base[]-field, or system_clock_reference_base[32..0] in MPEG-language, equals the number of ticks (mod 2^{33}) on the MPEG system clock, which runs at 90 kHz. The third invention claim in the introduction is to use the value of this SCR_base[] field in a sector/pack()together with the secret key K to determine whether or not this sector/pack() is to be 'summarized'. An example protocol to single out a sector for

'summarizing' is given in the flow-chart in Figure 3. The protocol also includes a way to exchange the value of the watermark/physical mark.

K is the aforementioned shared secret

- T_R is the SCR_base[] of the selected sector that is to be 'summarized'
- $C(T_I)$ is the digital summary of the selected sector
- *CGMS-D* are the two Copy-Generation Management System bits, denoting the copy state of the content (copy-free, copy-once, copy never).
- WM_i is (part) of the watermark payload that is to be shared with the drive
- $F()$ is a function which obfuscates its argument. The argument is transmitted to the drive in this form to keep the information secret and tamper-proof, from the "man in the middle". An example would be a one-way hash-function.

Typically, we would like to exchange a summary once per second, or 10 seconds. (In the example protocol of Figure 3, the length of this period determined in step (2) by monitoring the flipping of bit N of SCR_base[]. When N=16, as in the figure, the period is 2^{16+1} ticks ÷ 90,000 ticks/sec = 1.4 seconds. For N=17 we would 2×1.4 = 2.8 sec., for N=18 →5.6 sec etc.). Experiments with real DVD-video's suggest that in such a 1 sec period (and certainly in a 10 sec. period), there are ample sectors transmitted by the drive to allow the example algorithm to function properly. The algorithm of Figure 3 waits for 1.4 seconds, and then basically selects the K_0'th sector after that point, where K_0 is derived from the shared secret K.

Conclusions

- In a drive ↔ application protocol, drive and decoder need to verify that the content that they are transmitting viz. receiving is the same. This can be done by preferably securely exchanging summaries of the data bound for the application, and the data received by the application.
- For DVD-video, in implementing item 0, the drive and application cannot be absolutely certain which part of the data transmitted by the drive will be received by the decoder. To alleviate this problem the drive should only summarize sectors which start with the 4-byte pack_start_code. An improvement to avoid accidental false alarms through occurrence of pack_start_code in a non-video sector, is to also check that sector bytes 14...17 contain the so-called video_pack_start_code =0x000001E0. For *other* recording formats, the equivalent of pack_start_code should be chosen: i.e. a sequence of bits which (to a high

probability) is unique to a block of data that will be sent to the application, (as opposed to another destination within the PC).

- Because summarizing all sectors causes too much overhead (and is unnecessary from a security point of view) drive and decoder may just compute and *exchange a summary or characteristic of specific sectors* with pre-selected SCR_base[]. These sectors should be known only to drive and application. This selection of sectors could be made, based on a shared secret *K*. For *other* recording formats, the summary can likewise be computed based on (a characteristic of) a subset of the data transmitted from drive to the application. Selection of the subset should be based on the shared secret *K*.
- To avoid false alarms through a latency and delays in the communication between drive and application (beyond their control), both should store the last few summary-results against which they will verify incoming copy-protection messages.

A Man-in-the-Middle Attack

Consider the scenario in Figure 4. A PC with a copy protection compliant drive and compliant application (in this case an MPEG decoder card) is tricked into playing back an illegally copied disk with a watermarked film, by *pirated software* controlling drive and decoder. Obviously the pirated disk is without the proper diskmark. The software application controlling drive and decoder is pirated (downloaded from Internet etc.). The hack starts out with letting the drive and application authenticate each other, prior to playback. The drive sees no diskmark: this is not illegal in and of itself (a non copy-protected film on disk, or a legacy disk doesn't have a diskmark either).

When playback starts, the pirated control software, requests data sectors from the drive, and sends them to a *non-compliant decoder* (e.g. legacy existing software), whilst supplying other, pre-recorded data to the compliant decoder card, from, say, the hard-disk. The data from the disk-drive is watermarked, but this watermark is not recognized by the non-compliant decoder. The pre-recorded video from the hard-disk is *not* watermarked, so the compliant decoder doesn't see a watermark either. In this situation the compliant decoder will tell the compliant drive that it sees no watermarked video, so playback should continue; the drive hasn't seen a diskmark so it also decides that playback is legal.

The vulnerability exploited by this attack is obviously that the drive transfers data different from than that received by the decoder. To thwart the attack, in the *copy-control messages*,

1. the drive needs to report to the decoder a summary or characteristic of the video that it transmitted.
2. the decoder needs to report to the drive a summary or characteristic of the video that it received.

A complication in this scenario is that not all of the data requested from the drive is sent to the decoder: e.g. the table of contents of the disk and other file-management information is read and processed by the operating system, but not by the application! It is therefore essential that the summary that is being exchanged concerns *only* that part of the data that will end up in the application!

References:

1. ATAPI' working' specification from the MtFuji working group (expected to become the new SFFC/ANSI specification) MtFuji-3 Revision 0.94 (11th January 1999) ftp://ftp.avc-pioneer.com/Mtfuji3/Spec

2. Multi-Media Command Set (It only describes the "logical" interface (i.e. what commands to use to read CD-ROMs or to write DVD-RAMs etc.) MMC-2 Revision 9.0e (19th February 1999) http://www.symbios.com/t10

3. WG6 Media Identifier for Copyright Discussion Document by Philips - Jean-Paul Linnartz - Nat-Lab Oct. 30 1998

4. DVD Copy Protection System, An Overview, Joop Talstra, Ton Kalker, Jean Paul Linnartz (Philips Research) and Mark Hollar, Patrice Capitant (Macrovision), Jan. 22 1999.

5. Overview of CPS, Philips-Macrovision-Digimarc, G. Depovere, T. Kalker, J-P. Linnartz, J. Talstra, Nat-Lab

6. Applied Cryptography, by Bruce Schneier, (John Wiley, NY, 1997).

7. MPEG-2 standard, Part 1, System (ISO/IEC13818-1).

CLAIMS:

1. A method of copy protection substantially as described herein with reference to one or more of the accompanying drawings.

2. A method of exchanging copy protection information regarding an information carrying medium substantially as described herein with reference to one or more of the accompanying drawings.

3. A copy protection system substantially as described herein with reference to one or more of the accompanying drawings.

4. A device substantially as described herein with reference to one or more of the accompanying drawings.

5. A device wherein a method as claimed in Claim 1 or 2 is used for copy protecting the content stored in the device.

ABSTRACT:

A method is presented to allow a reader-unit (e.g. a DVD drive) and an application unit (e.g. an MPEG decoder) to exchange copy-protection information regarding the information carrying medium (disk) and the content on that medium. The method is cryptographically secure, taking into account the situation where reader-unit and application-unit are connected to an *open* bus in a Personal Computer. In view of the high-volume nature of the drive, the method can be implemented cheaply. The method is robust against a so-called man-in-the-middle attack.

Keywords:
Copy-protection systems, watermarking, optical disk recording, cryptography, play-control.

Fig. 3

PHP 17.410

Play-Control Set up

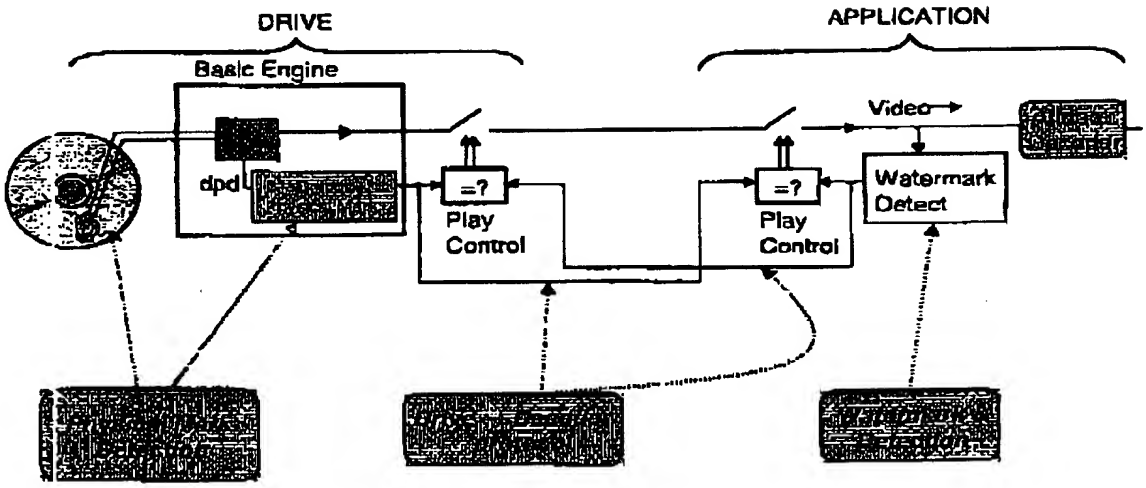

Figure 1 Schematic of play control during playback for video in a player. In this set up, drive and application have their own play-control unit, e.g. because they are physically separated—like in a PC. Nevertheless, they still rely on each other for information to enact proper play control.

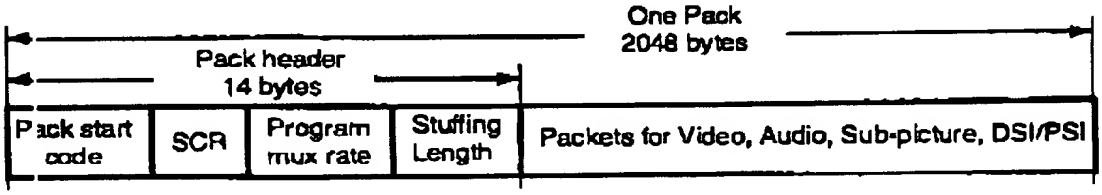

Figure 2 **Outline of a DVD-Video pack ( )**

PHN 17.410

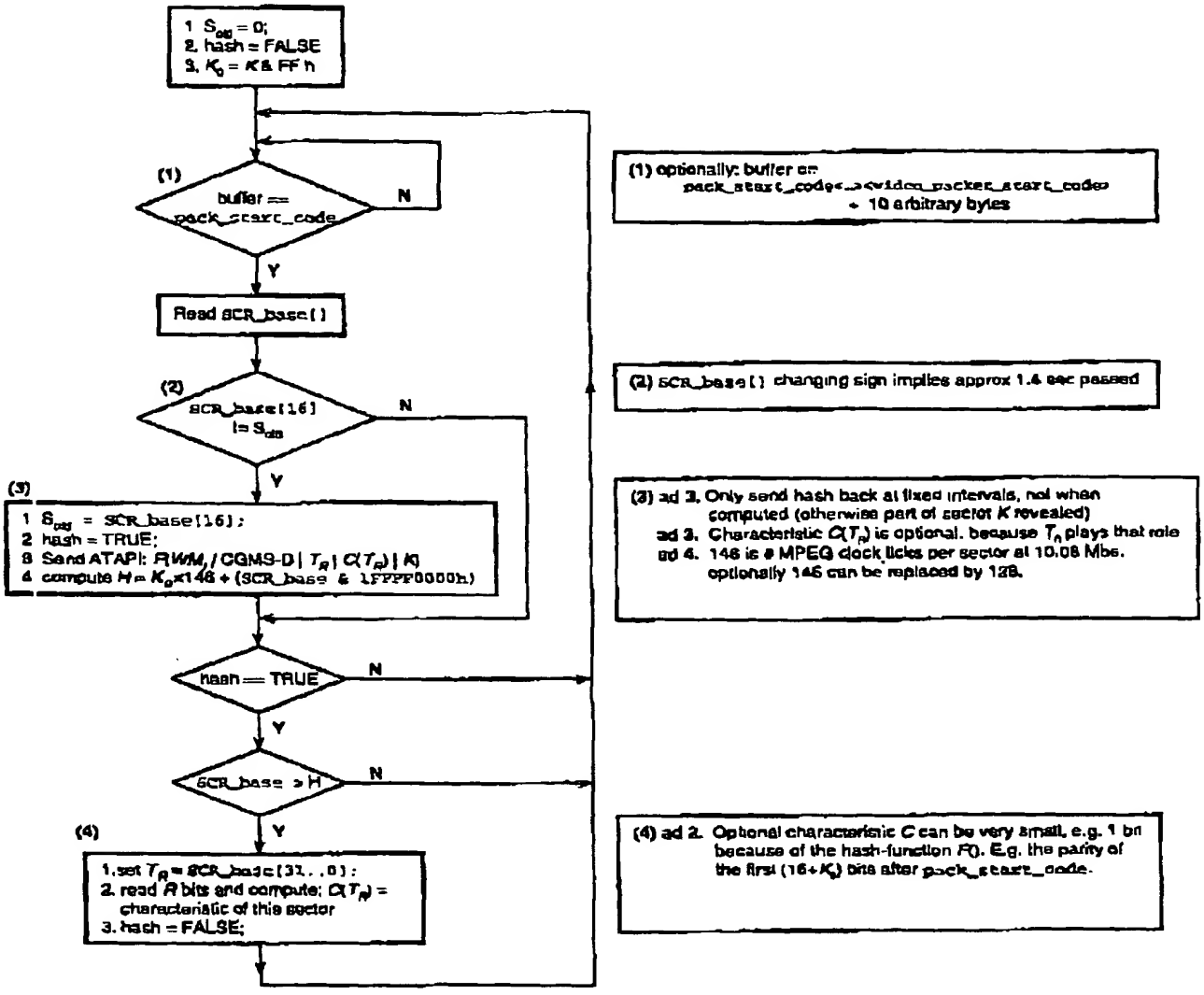

Figure 3 Computation of the summary at the decoder for transmitting the watermark payload securely to the drive. The drive goes through the same algorithm, but in step (3) replaces watermark payload WM_i by W, the payload value stored in the diskmark. Note: in item (3) 4., H and K_0 essentially select the K_0-th sector after the beginning of the period signalled by `SCR_base[16]` flipping.

PHN 17.410

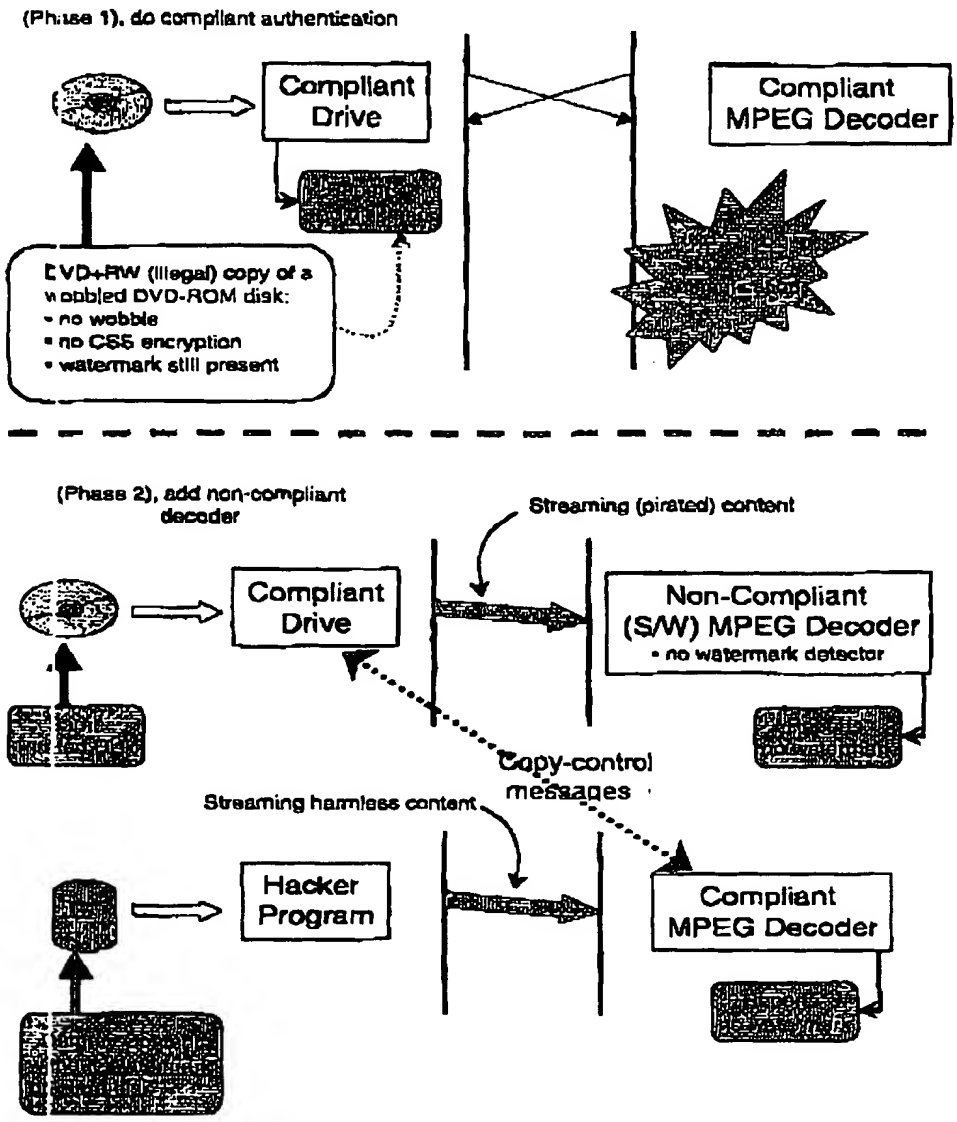

Figure 4 It is necessary that both drive and compliant decoder/watermark-detector are talking about the same content. If not, we can play back a pirated disk (no proper disk mark, but watermarked content) by splicing in dummy, non-watermarked content to the compliant decoder after the authentication phase.

PHN 17.410

Table 1 DVD Program Stream pack ( ) header

Field	Number of bits	Value
pack_start_code	32	000001BA h
'01'	2	01 b
SCR_base[32..30]	3	SCR_base[32] = 0
marker_bit	1	1 b
SCR_base[29..15]	15	
marker_bit	1	1 b
SCR_base[14..0]	15	
marker_bit	1	1 b
SCR_extension	9	
marker_bit	1	1 b
program_mux_rate	22	0189C3 x
marker_bit	1	1 b
marker_bit	1	1 b
Reserved	5	11111 b
pack_stuffing_length	3	

Fig. 5